# Mkstemp

Don't be complacent. Several vulnerabilities possible.

Sean Barnum, Cigital, Inc. [vita[1]]

Copyright © 2007 Cigital, Inc.

2007-04-02

## Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5650 bytes

| | |
|---|---|
| **Attack Category** | • Malicious Input<br>• Privilege Exploitation<br>• File Manipulation |
| **Vulnerability Category** | • Temporary file creation problem<br>• Privilege escalation problem |
| **Software Context** | • File Creation |
| **Location** | • stdlib.h |
| **Description** | Use of mkstemp() to create a temporary file should not lead to complacency, as it is still possible for vulnerabilities to be present.<br><br>The mkstemp() function generates a unique temporary file name from the supplied template, opens a file of that name using the O_EXCL flag (guaranteeing the current process to be the only user) and returns a file descriptor.<br><br>The POSIX specification does not say anything about file modes, so the application should make sure its umask is set appropriately before calling mkstemp.<br><br>mkstemp() is designed to facilitate the creation of a temporary file in a way that is more secure than the use of mktemp() followed by open(). Use of mkstemp() does avoid the class of race conditions that involves a third party guessing the temporary file name and creating that file between the mktemp() and open() calls. However, use of mkstemp() does not eliminate all vulnerabilities. While use of the returned file descriptor is safe, manipulation of the temporary file by name can introduce other vulnerabilties. |
| **APIs** | **Function Name** \| **Comments**<br>mkstemp \| |
| **Method of Attack** | A common practice of installing 'tmpwatch' utility or similar software configured to sweep |

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

| | |
|---|---|
| | the /tmp directory on Linux and UNIX systems can compromise secure temporary file creation mechanisms in certain applications, creating a potential privilege escalation scenario.<br><br>By taking advantage of the operation of the cleanup utility, and exploiting either an "unlink" race condition or a deletion that occurs because a process that created a temporary file is suspended for an extended period of time, an attacker can potentially substitute an imposter file for the original temporary file. This can result in a privilege escalation scenario. |
| **Exception Criteria** | The mkstemp() function is safe if only the descriptor is used and the returned filename is not used in a subsequent function call with extra privileges. |
| **Solutions** | |
| **Signature Details** | int mkstemp(char *template); |

| **Solution Applicability** | **Solution Description** | **Solution Efficacy** |
|---|---|---|
| mkstemp() is used by privileged programs in a system with a temporary file cleanup utility. | If possible, avoid using the name of the temporary file to perform any sensitive file operations.<br><br>Using a file cleanup utility with more secure logic may somewhat reduce vulnerability. But it is not clear that there is a design that eliminates all risk.<br><br>Privileged applications should use private temporary directories for sensitive files, if possible. (Doing this in a mandatory fashion may, however, create portability issues.) | Efficacy depends on particular solution. |

| | |
|---|---|
| **Examples of Incorrect Code** | <pre>int fd = mkstemp("FooTemplate");<br>// write to fd<br>const char *fileName =<br>useFstatToGetFileNameFromDescriptor(fd);<br>[...]<br>// attacker could potentially<br>substitute a different file to be<br>opened in the next line<br>int fd2 = open(fileName, O_EXCL);<br>// rely on data read from fd2</pre> |
| **Examples of Corrected Code** | <pre>// set umask appropriately<br>int fd = mkstemp("FooTemplate");<br>// write to fd<br>// rely on data read from fd</pre> |
| **Source References** | • http://www.bindview.com/Services/Razor/Papers/2002/mkstemp.cfm<br>• ITS4 Source Code Vulnerability Scanning Tool[3] |
| **Recommended Resources** | • Man page for functions to create a temporary file[4]<br>• Man page for mkstemp[5] |
| **Discriminant Set** | **Operating System**: • UNIX (All)<br>**Languages**: • C • C++ |

## Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com[1].

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com